Realisability Semantics for Intersection Types and Expansion Variables

Fairouz Kamareddine, Karim Nour, Vincent Rahli, J. B. Wells

Realisability Semantics for Intersection Types and Expansion Variables

Fairouz Kamareddine, Karim Nour, Vincent Rahli and J. B. Wells

http://www.macs.hw.ac.uk/ultra/

May 13, 2009

Abstract

Expansion was invented at the end of the 1970s for calculating *principal typings* for λ-terms in type systems with *intersection types*. *Expansion variables* (E-variables) were invented at the end of the 1990s to simplify and help mechanise expansion. Recently, E-variables have been further simplified and generalised to also allow calculating type operators other than just intersection. There has been much work on denotational semantics for type systems with intersection types, but none whatsoever before now on type systems with E-variables. Building a semantics for E-variables turns out to be challenging. To simplify the problem, we consider only E-variables, and not the corresponding operation of expansion. We develop a realisability semantics where each use of an E-variable in a type corresponds to an independent degree at which evaluation occurs in the λ-term that is assigned the type. In the λ-term being evaluated, the only interaction possible between portions at different degrees is that higher degree portions can be passed around but never applied to lower degree portions. We apply this semantics to two intersection type systems. We show these systems are sound, that completeness does not hold for the first system, and completeness holds for the second system when only one E-variable is allowed (although it can be used many times and nested). As far as we know, this is the first study of a denotational semantics of intersection type systems with E-variables (using realisability or any other approach).

1 Introduction

Intersection types were developed in the late 1970s to type λ-terms that are untypable with simple types; they do this by providing a kind of finitary type polymorphism where the usage of types is listed rather than quantified over. They have been useful in reasoning about the semantics of the λ-calculus, and have been investigated for use in static program analysis. Coppo, Dezani, and Venneri [5] introduced the operation of *expansion* on *typings* (pairs of a type environment and a result type) for calculating the possible typings of a term when using intersection types. Expansion is a crucial part of a procedure for calculating *principal typings* and thus helps support compositional type inference. As a simple example, the λ-term $M = (\lambda x.x(\lambda y.yz))$ can be assigned the typing $\Phi_1 = \langle (z : a) \vdash (((a \to b) \to b) \to c) \to c \rangle$, which happens to be its principal typing. The term M can also be assigned the typing $\Phi_2 = \langle (z : a_1 \sqcap a_2) \vdash (((a_1 \to b_1) \to b_1) \sqcap ((a_2 \to b_2) \to b_2) \to c) \to c \rangle$, and an expansion operation can obtain Φ_2 from Φ_1. Because the early definitions of expansion were complicated, E-variables were introduced in order to make the calculations easier to mechanise and reason about. For example, in System E [3], the typing Φ_1 from above is replaced by $\Phi_3 = \langle (z : ea) \vdash (e((a \to b) \to b) \to c) \to c \rangle$, which differs from Φ_1 by the insertion of the E-variable e at two places, and Φ_2 can be obtained from Φ_3 by substituting for e the *expansion term* $E = (a := a_1, b := b_1) \sqcap (a := a_2, b := b_2)$. Carlier and Wells [4] have surveyed the history of expansion and also E-variables.

Various kinds of denotational semantics have helped in reasoning about the properties of entire type systems and also of specific typed terms. E-variables pose serious challenges for semantics. Most commonly, a type's semantics is given as a set of closed λ-terms with behaviour related to the specification given by the type. In many kinds of semantics, the meaning of a type T is calculated by an expression $[T]_\nu$ that takes two parameters, the type T and also a valuation ν that assigns to type variables the same kind of meanings that are assigned to types. To extend this idea to types with E-variables, we would need to devise some space of possible meanings for E-variables. Given that a type eT can be turned by expansion into a new type $S_1(T) \sqcap S_2(T)$, where S_1 and S_2 are arbitrary substitutions (in fact, they

can be arbitrary further expansions), and that this can introduce an unbounded number of new variables (both E-variables and regular type variables), the situation is complicated.

Because it is unclear how to devise a space of meanings for expansions and E-variables, we instead develop a space of meanings for types that is hierarchical in the sense of having many degrees. When assigning meanings to types, we make each use of E-variables simply change degrees. We specifically avoid trying to give a semantics to the operation of expansion, and instead treat only the E-variables. Although this idea is not perfect, it seems to go quite far in giving an intuition for E-variables, namely that each E-variable acts as a kind of capsule that isolates parts of the λ-term being analysed by the typing. Parts of the λ-term that are typed inside the uses of the E-variable-introduction typing rule for a particular E-variable e can interact with each other, and parts outside e can only pass the parts inside e around. The E-variable e of course also shows up in the types, and isolates the portions of the types contributed by the portions of the term inside the corresponding uses of E-variable-introduction.

The semantic approach we use is *realisability semantics*. Atomic types are interpreted as sets of λ-terms that are *saturated*, meaning that they are closed under β-expansion (i.e., β-reduction in reverse). Arrow and intersection types are interpreted naturally by function spaces and set intersection. Realisability allows showing *soundness* in the sense that the meaning of a type T contains all closed λ-terms that can be assigned T as their result type. This has been shown useful in previous work for characterising the behaviour of typed λ-terms [14]. One also wants to show *completeness* (the converse of soundness), i.e., that every closed λ-term in the meaning of T can be assigned T as its result type.

Hindley [10, 11, 12] was the first to study completeness for a simple type system. Then, he generalised his completeness proof for an intersection type system [9]. Using his completeness result for the realisability semantics based on the sets of λ-terms saturated by β-equivalence, Hindley has shown that simple types are uniquely realised by the λ-terms that are typable by these types in a type system similar to $\lambda_{\rightarrow}$ [2] augmented with a β-equivalence rule (this rule assigns the same typings to β-equivalent terms) [10]. He proved this result using saturation by $\beta\eta$-equivalence w.r.t. a type system similar to $\lambda_{\rightarrow}$ augmented with a $\beta\eta$-equivalence rule too. Hindley also established completeness using saturation by β-equivalence for his intersection type system [9]. In this paper, our completeness result depends instead only on a weaker notion than β-equivalence (saturation by β-expansion).

Other work on realisability we consulted includes that by Labib-Sami [15], Farkh and Nour [7], and Coquand [6], although none of this work deals with intersection types or E-variables. Related work on realisability that deals with intersection types includes that by Kamareddine and Nour [13], which gives a realisability semantics with soundness and completeness for an intersection type system. The system of Kamareddine and Nour is different from those in this paper, because it allows the universal type ω. We do not know how to build a semantics that supports both ω and E-variables. The method of degrees we use in this paper would need to assign ω to every degree, which is impossible. Further work is needed on this point.

In this paper we study the λI-calculus typed with two representative intersection type systems. The restriction to λI (where in $\lambda x.M$, the variable x must be free in M) is motivated by not knowing how to support the ω type. For one of these systems, we show that subject reduction (SR) and hence completeness do not hold whereas for the second system, SR holds and completeness will hold if at most one E-variable is used (although this E-variable may be used in many places and also nested). This is the first paper that studies denotational semantics of intersection type systems with E-variables, using realisability or any other approach. One of our contributions is to outline the difficulties of doing so.

The semantics we build in this paper, defines sets of realisers (functions/programs satisfying the requirements of some specification) of types. Such a model can help to highlight the relation between typable terms of the untyped lambda-calculus and types w.r.t. a type system. Interpreting types in a model helps to understand the meaning of a type (w.r.t. the model) which is defined as a purely syntactic form and is clearly used as a meaningful expression. For example, the integer type, whatever its notation is, is always used as the type of each integer. In the open problems published in the

proceedings of the Lecture Notes in Computer Science symposium help in 1975 [8], it is suggested that an arrow type expresses functionality. In that way, models based on term-models have been built for intersection type systems [9, 13]. In these works, intersection types (introduced to be able to type more terms than in the Simply Typed Lambda Calculus) are interpreted by set-theoretical intersection of meanings. Even if expansion variables have been introduced to give a simple formalisation of the expansion mechanism, i.e., as a syntactic object, we are interested in the meaning of such a syntactic object. We are particularly interested by answering these questions: What does an expansion variable applied to a type stand for? What are the realisers of such a type? How can the relation between terms and types w.r.t. a type system be described? How can we extend models such as the one built by Kamareddine and Nour [13] to a type system with expansion?

Section 2 introduces the $\lambda I^{\mathbb{N}}$-calculus, which is the λI-calculus with each variable marked by a natural number *degree*. Section 3 introduces the syntax and terminology for types, and also the realisability semantics. Section 4 introduces our two intersection type systems with E-variables. In one system, the syntax of types is not restricted but in the other system it is restricted but then extended with a subtyping relation. We show that SR and completeness do not hold for the first system, and that SR holds for the second system. We also show the soundness of the realisability semantics for both systems and give a number of examples. Section 5 shows completeness does not hold for the second system if more than one expansion variable is used, but does hold for a restriction of this system to one single E-variable (which can be used in many places and also nested). This is an important study in the semantics of intersection type systems with expansion variables. Section 6 concludes. Full proofs can be downloaded from the web page of the authors as well as further results that include strong normalisation of the typable terms and the relation to the usual unindexed λI-calculus.

2 The pure $\lambda I^{\mathbb{N}}$-calculus

In this section we give $\lambda I^{\mathbb{N}}$, an indexed version of the λI-calculus where indices (which range over the set of natural numbers $\mathbb{N} = \{0, 1, 2, \ldots\}$) help categorise the *good terms* where the degree of a function is never larger than that of its argument. This amounts to having the full λI-calculus at each degree (index) and creating new λI-terms through a mixing recipe. Let n, m be metavariables which range over the set of natural numbers $\mathbb{N}$. We assume that if a metavariable v ranges over a set $\mathcal{S}$ then v_i for $i \geq 0$ and v', v'', etc. also range over $\mathcal{S}$. A binary relation is a set of pairs. Let rel range over binary relations. Let $\mathrm{dom}(rel) = \{x \mid \langle x, y\rangle \in rel\}$ and $\mathrm{ran}(rel) = \{y \mid \langle x, y\rangle \in rel\}$. A function is a binary relation fun such that if $\{\langle x, y\rangle, \langle x, z\rangle\} \subseteq fun$ then $y = z$. Let fun range over functions. Let $s \to s' = \{fun \mid \mathrm{dom}(fun) \subseteq s \wedge \mathrm{ran}(fun) \subseteq s'\}$. We sometimes write $x : s$ instead of $x \in s$.

Definition 1

i) Let $\mathcal{V}$ be a denumerably infinite set of variables. The set of terms $\mathcal{M}$, the set of good terms $\mathbb{M} \subset \mathcal{M}$, the set of free variables $FV(M)$ of $M \in \mathcal{M}$, the degree $\mathrm{d}(M)$ of a term M and the joinability $M \diamond N$ of terms M and N (which ensures that in any term, each variable has a unique degree) are defined by simultaneous induction:

- If $x \in \mathcal{V}$, $n \in \mathbb{N}$, then $x^n \in \mathcal{M} \cap \mathbb{M}$, $FV(x^n) = \{x^n\}$, and $\mathrm{d}(x^n) = n$.
- If $M, N \in \mathcal{M}$ such that $M \diamond N$ (see below), then
  - $(M\ N) \in \mathcal{M}$, $FV((M\ N)) = FV(M) \cup FV(N)$ and $\mathrm{d}((M\ N)) = \min(\mathrm{d}(M), \mathrm{d}(N))$ (where $\min$ is the minimum)
  - If $M \in \mathbb{M}$, $N \in \mathbb{M}$ and $\mathrm{d}(M) \leq \mathrm{d}(N)$ then $(M\ N) \in \mathbb{M}$.
- If $M \in \mathcal{M}$ and $x^n \in FV(M)$, then
  - $(\lambda x^n.M) \in \mathcal{M}$, $FV((\lambda x^n.M)) = FV(M) \setminus \{x^n\}$, and $\mathrm{d}((\lambda x^n.M_1)) = \mathrm{d}(M_1)$.
  - If $M \in \mathbb{M}$ then $\lambda x^n.M \in \mathbb{M}$.

ii) Let $M, N \in \mathcal{M}$. We say that M and N are joinable and write $M \diamond N$ iff $\forall x \in \mathcal{V}$, if $x^m \in FV(M)$

and $x^n \in FV(N)$, then $m = n$. If $\mathcal{X} \subseteq \mathcal{M}$ such that $\forall M, N \in \mathcal{X}, M \diamond N$, we write, $\diamond\mathcal{X}$. If $\mathcal{X} \subseteq \mathcal{M}$ and $M \in \mathcal{M}$ such that $\forall N \in \mathcal{X}, M \diamond N$, we write, $M \diamond \mathcal{X}$.

iii) We adopt the usual definition [1, 14] of subterms and the convention for parentheses and their omission. Note that a subterm of $M \in \mathcal{M}$ (resp. $\mathbb{M}$) is also in $\mathcal{M}$ (resp. $\mathbb{M}$). We let x, y, z, etc. range over $\mathcal{V}$ and M, N, P, etc. range over $\mathcal{M}$ and use $=$ for syntactic equality.

iv) For each $n \in \mathbb{N}$, we let: • $\mathcal{M}^n = \{M \in \mathcal{M} \mid \mathrm{d}(M) = n\}$
• $\mathcal{M}^{>n} = \mathcal{M}^{\geq n+1}$ • $\mathcal{M}^{\geq n} = \{M \in \mathcal{M} \mid \mathrm{d}(M) \geq n\}$ • $\mathbb{M}^n = \mathbb{M} \cap \mathcal{M}^n$

v) For $m \geq 0$, $M[(x_i^{n_i} := N_i)_{1 \leq i \leq m}]$ (or simply $M[(x_i^{n_i} := N_i)_m]$), the simultaneous substitution of N_i for all free occurrences of $x_i^{n_i}$ in M only matters when $\diamond\mathcal{X}$ where $\mathcal{X} = \{M\} \cup \{N_i \mid 1 \leq i \leq m\} \subseteq \mathcal{M}$. Hence we restrict substitution accordingly to incorporate the $\diamond$ condition. With $\mathcal{X}$ as above, $M[(x_i^{n_i} := N_i)_m]$ is only defined when $\diamond\mathcal{X}$. We write $M[(x_i^{n_i} := N_i)_{1 \leq i \leq 1}]$ as $M[x_1^{n_1} := N_1]$.

vi) We take terms modulo α*-conversion* given by: $\lambda x^n.M = \lambda y^n.(M[x^n := y^n])$ where $\forall m, y^m \notin FV(M)$. We use the Barendregt convention (BC) where the names of bound variables differ from the free ones and where we rewrite terms so that not both λx^n and λx^m co-occur when $n \neq m$.

vii) A relation R on $\mathcal{M}$ is *compatible* iff for all $M, N, P \in \mathcal{M}$:
- If $\langle M, N\rangle \in R$ and $x^n \in FV(M) \cap FV(N)$ then $\langle \lambda x^n.M, \lambda x^n.N\rangle \in R$.
- If $\langle M, N\rangle \in R$, $M \diamond P$ and $N \diamond P$ then $\langle MP, NP\rangle \in R$ and $\langle PM, PN\rangle \in R$.

viii) The reduction relation $\rhd_\beta$ on $\mathcal{M}$ is defined as the least compatible relation closed under the rule: $(\lambda x^n.M)N \rhd_\beta M[x^n := N]$ if $\mathrm{d}(N) = n$.

ix) We denote by $\rhd_\beta^*$ the reflexive and transitive closure of $\rhd_\beta$. We denote by $\simeq_\beta$ the equivalence relation induced by $\rhd_\beta^*$.

Beta reduction is well defined on the $\lambda I^{\mathbb{N}}$-calculus, i.e., if $M \in \mathcal{M}$ and $M \rhd_\beta N$ then $N \in \mathcal{M}$. (Note that because $\mathrm{d}(x^0) = 0 \neq 1 = \mathrm{d}(z^1)$, then $(\lambda x^0.x^0 y^0)z^1 \not\rhd_\beta z^1 y^0$.) Hence, $\rhd_\beta^*$ is also well defined on $\mathcal{M}$. Beta reduction preserves the free variables, degrees and goodness of terms, i.e., if $M \rhd_\beta^* N$ then $FV(M) = FV(N)$, $\mathrm{d}(M) = \mathrm{d}(N)$ and M is good iff N is good.

The next definition turns terms of degree n into terms of higher degrees and also, if $n > 0$, they can be turned into terms of lower degrees. Note that $^+$ and $^-$ are well behaved operations with respect to all that matters (free variables, reduction, joinability, substitution, etc.).

Definition 2

i) We define $^+ : \mathcal{M} \to \mathcal{M}$ and $^- : \mathcal{M}^{>0} \to \mathcal{M}$ by:
• $(x^n)^+ = x^{n+1}$ • $(M_1\ M_2)^+ = M_1^+\ M_2^+$ • $(\lambda x^n.M)^+ = \lambda x^{n+1}.M^+$
• $(x^n)^- = x^{n-1}$ • $(M_1\ M_2)^- = M_1^-\ M_2^-$ • $(\lambda x^n.M)^- = \lambda x^{n-1}.M^-$

ii) Let $\mathcal{X} \subseteq \mathcal{M}$. If $\forall M \in \mathcal{X}$, $\mathrm{d}(M) > 0$, we write $\mathrm{d}(\mathcal{X}) > 0$. We define:
• $\mathcal{X}^+ = \{M^+ \mid M \in \mathcal{X}\}$ • If $\mathrm{d}(\mathcal{X}) > 0$, $\mathcal{X}^- = \{M^- \mid M \in \mathcal{X}\}$.

iii) We define M^{-n} by induction on $\mathrm{d}(M) \geq n \geq 0$. If $n = 0$ then $M^{-n} = M$ and if $n \geq 0$ then $M^{-(n+1)} = (M^{-n})^-$.

3 The types and their realisability semantics

This paper studies two type systems. In the first, there are no restrictions on where the arrow occurs. In the second, arrows cannot occur to the left of intersections or expansions. The next definition gives these two basic sets of types and the notions of a degree of a type and of a good type.

Definition 3 (TYPES, GOOD TYPES, DEGREE OF A TYPE)

i) Assume two denumerably infinite sets $\mathcal{A}$ (atomic types) and $\mathcal{E}$ (expansion variables). Let a, b, c, etc. range over $\mathcal{A}$ and e range over $\mathcal{E}$.

ii) The sets of types $\mathcal{T}$, $\mathbb{U}$ and $\mathbb{T}$ are defined by $\mathcal{T} ::= \mathcal{A} \mid \mathcal{T} \to \mathcal{T} \mid \mathcal{T} \sqcap \mathcal{T} \mid \mathcal{E}\mathcal{T}$ and $\mathbb{U} ::= \mathbb{U} \sqcap \mathbb{U} \mid \mathcal{E}\mathbb{U} \mid \mathbb{T}$ where $\mathbb{T} ::= \mathcal{A} \mid \mathbb{U} \to \mathbb{T}$ (note that $\mathbb{T}$ and $\mathbb{U}$ are defined simultaneously). Note that $\mathbb{T} \subseteq \mathbb{U} \subseteq \mathcal{T}$. We let T, U, V, W (resp. T, resp. U, V, W) range over $\mathcal{T}$ (resp. $\mathbb{T}$, resp. $\mathbb{U}$). We quotient types by taking $\sqcap$ to be commutative, associative, idempotent, and to satisfy $e(U_1 \sqcap U_2) = eU_1 \sqcap eU_2$.

iii) Denote $e_{i_l} \dots e_{i_n}$ by $\vec{e}_{i(l:n)}$ and $U_n \sqcap U_{n+1} \dots \sqcap U_m$ by $\sqcap_{i=n}^{m} U_i$ ($n \leq m$).

iv) We define a function $\mathrm{d} : \mathcal{T} \to \mathbb{N}$ by (hence d is also defined on $\mathbb{U}$):
- $\mathrm{d}(a) = 0$
- $\mathrm{d}(U \to T) = \min(\mathrm{d}(U), \mathrm{d}(T))$
- $\mathrm{d}(eU) = \mathrm{d}(U) + 1$
- $\mathrm{d}(U \sqcap V) = \min(\mathrm{d}(U), \mathrm{d}(V))$.

v) We define the good types on $\mathcal{T}$ by (this also defines good types on $\mathbb{U}$):
- If $a \in \mathcal{A}$, then a is good
- If U is good and $e \in \mathcal{E}$, then eU is good
- If U, T are good and $\mathrm{d}(U) \geq \mathrm{d}(T)$, then $U \to T$ is good
- If U, V are good and $\mathrm{d}(U) = \mathrm{d}(V)$, then $U \sqcap V$ is good

Definition 4 (ENVIRONMENTS) *i*) A type environment is a set $\{x_i^{n_i} : U_i \mid 1 \leq i \leq n$ where $n \geq 0$ and $\forall 1 \leq i, j \leq n,$ if $i \neq j$ then $x_i^{n_i} \neq x_j^{n_j}\}$. We denote such environment (call it Γ) by $x_1^{n_1} : U_1, x_2^{n_2} : U_2, \dots, x_n^{n_n} : U_n$ or simply by $(x_i^{n_i} : U_i)_n$ and define $dom(\Gamma) = \{x_i^{n_i} \mid 1 \leq i \leq n\}$. We use Γ, Δ to range over environments and write $()$ for the empty environment.
Of course on $\mathcal{T}$, type environments take variables in $\mathcal{V}$ to $\mathcal{T}$. On $\mathbb{U}$, they take variables in $\mathcal{V}$ to $\mathbb{U}$.
- We say that Γ is good iff , for every $1 \leq i \leq k$, U_i is good.
- We say that $\mathrm{d}(\Gamma) > 0$ iff for every $1 \leq i \leq k$, $\mathrm{d}(U_i) > 0$ and $n_i > 0$.

ii) If $\Gamma = (x_i^{n_i} : U_i)_n$ and $x^m \notin dom(\Gamma)$, then we write $\Gamma, x^m : U$ for the type environment $x_1^{n_1} : U_1, \dots, x_n^{n_n} : U_n, x^m : U$.

iii) Let $\Gamma_1 = (x_i^{n_i} : U_i)_n, (y_j^{m_j} : V_j)_m$ and $\Gamma_2 = (x_i^{n_i} : U_i')_n, (z_k^{r_k} : W_k)_r$. We write $\Gamma_1 \sqcap \Gamma_2$ for the type environment $(x_i^{n_i} : U_i \sqcap U_i')_n, (y_j^{m_j} : V_j)_m, (z_k^{r_k} : W_k)_r$. Note that $dom(\Gamma_1 \sqcap \Gamma_2) = dom(\Gamma_1) \cup dom(\Gamma_2)$ and that $\sqcap$ is commutative, associative and idempotent on environments.

iv) $e\Gamma = (x_i^{n_i+1} : eT_i)_n$ where $\Gamma = (x_i^{n_i} : T_i)_n$. So $e(\Gamma_1 \sqcap \Gamma_2) = e\Gamma_1 \sqcap e\Gamma_2$.

v) We say that Γ_1 is joinable with Γ_2 and write $\Gamma_1 \diamond \Gamma_2$ iff
$\forall x \in \mathcal{V}$, if $x^m \in dom(\Gamma_1)$ and $x^n \in dom(\Gamma_2)$, then $m = n$.

Definition 5 (DEGREE DECREASING OF A TYPE) *i*) If $\mathrm{d}(U) > 0$, we inductively define the type U^- by:
- $(U_1 \sqcap U_2)^- = U_1^- \sqcap U_2^-$
- $(eU)^- = U$

If $\mathrm{d}(U) \geq n \geq 0$, U^{-n} is defined as for M^{-n} in definition 2.

ii) If $\Gamma = (x_i^{n_i} : U_i)_k$ and $\mathrm{d}(\Gamma) > 0$, then we let $\Gamma^- = (x_i^{n_i - 1} : U_i^-)_k$.
If $\mathrm{d}(\Gamma) \geq n \geq 0$, Γ^{-n} is defined as for M^{-n} in definition 2.

iii) If U is a type and Γ is a type environment such that $\mathrm{d}(\Gamma) > 0$ and $\mathrm{d}(U) > 0$, then we let $(\langle \Gamma \vdash_2 U \rangle)^- = (\langle \Gamma^- \vdash_2 U^- \rangle)$.

Saturated sets and the interpretations and meanings of types are crucial to a realisability semantics:

Definition 6 (SATURATED SETS) Let $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{M}$.

i) We use $\mathcal{P}(\mathcal{X})$ to denote the powerset of $\mathcal{X}$, i.e. $\{\mathcal{Y} \mid \mathcal{Y} \subseteq \mathcal{X}\}$.

ii) We let $\mathcal{X} \rightsquigarrow \mathcal{Y} = \{M \in \mathcal{M} \mid \forall N \in \mathcal{X}$, if $M \diamond N$ then $M\ N \in \mathcal{Y}\}$.

iii) $\mathcal{X}$ is saturated iff whenever $M \rhd_\beta^* N$ and $N \in \mathcal{X}$, then $M \in \mathcal{X}$.

Definition 7 (INTERPRETATIONS AND MEANING OF TYPES) Let $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$ where $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ and $\mathcal{V}_1, \mathcal{V}_2$ are both denumerably infinite.

i) Let $x \in \mathcal{V}_1$ and $n \in \mathbb{N}$. We define $\mathcal{N}_x^n = \{x^n\ N_1 ... N_k \in \mathbb{M} \mid k \geq 0\}$.

ii) An interpretation $\mathcal{I} : \mathcal{A} \to \mathcal{P}(\mathcal{M}^0)$ is a function such that for all $a \in \mathcal{A}$:
- $\mathcal{I}(a)$ is saturated and
- $\forall x \in \mathcal{V}_1,\ \mathcal{N}_x^0 \subseteq \mathcal{I}(a) \subseteq \mathbb{M}^0$.

$$\frac{T \text{ good} \quad \mathrm{d}(T) = n}{x^n : \langle (x^n : T) \vdash_1 T \rangle} \ (ax)$$

$$\frac{T \text{ good}}{x^0 : \langle (x^0 : T) \vdash_2 T \rangle} \ (ax)$$

$$\frac{M : \langle \Gamma, (x^n : U) \vdash_i T \rangle}{\lambda x^n.M : \langle \Gamma \vdash_i U \to T \rangle} \ (\to_I)$$

$$\frac{M_1 : \langle \Gamma_1 \vdash_i U \to T \rangle \quad M_2 : \langle \Gamma_2 \vdash_i U \rangle \quad \Gamma_1 \diamond \Gamma_2}{M_1 M_2 : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_i T \rangle} \ (\to_E)$$

$$\frac{M : \langle \Gamma_1 \vdash_i U_1 \rangle \quad M : \langle \Gamma_2 \vdash_i U_2 \rangle}{M : \langle \Gamma_1 \sqcap \Gamma_2 \vdash_i U_1 \sqcap U_2 \rangle} \ (\sqcap)$$

$$\frac{M : \langle \Gamma \vdash_i U \rangle}{M^+ : \langle e\Gamma \vdash_i eU \rangle} \ (exp)$$

$$\frac{M : \langle \Gamma \vdash_2 U \rangle \quad \langle \Gamma \vdash_2 U \rangle \sqsubseteq \langle \Gamma' \vdash_2 U' \rangle}{M : \langle \Gamma' \vdash_2 U' \rangle} \ (\sqsubseteq)$$

$$\frac{}{\Phi \sqsubseteq \Phi} \ (ref)$$

$$\frac{\Phi_1 \sqsubseteq \Phi_2 \quad \Phi_2 \sqsubseteq \Phi_3}{\Phi_1 \sqsubseteq \Phi_3} \ (tr)$$

$$\frac{U_2 \text{ good} \quad \mathrm{d}(U_1) = \mathrm{d}(U_2)}{U_1 \sqcap U_2 \sqsubseteq U_1} \ (\sqcap_e)$$

$$\frac{U_1 \sqsubseteq V_1 \quad U_2 \sqsubseteq V_2}{U_1 \sqcap U_2 \sqsubseteq V_1 \sqcap V_2} \ (\sqcap)$$

$$\frac{U_2 \sqsubseteq U_1 \quad T_1 \sqsubseteq T_2}{U_1 \to T_1 \sqsubseteq U_2 \to T_2} \ (\to)$$

$$\frac{U_1 \sqsubseteq U_2}{eU_1 \sqsubseteq eU_2} \ (\sqsubseteq_{exp})$$

$$\frac{U_1 \sqsubseteq U_2}{\Gamma, (y^n : U_1) \sqsubseteq \Gamma, (y^n : U_2)} \ (\sqsubseteq_c)$$

$$\frac{U_1 \sqsubseteq U_2 \quad \Gamma_2 \sqsubseteq \Gamma_1}{\langle \Gamma_1 \vdash_2 U_1 \rangle \sqsubseteq \langle \Gamma_2 \vdash_2 U_2 \rangle} \ (\sqsubseteq_{\langle\rangle})$$

Figure 1: Typing rules / Subtyping rules

iii) Let an interpretation $\mathcal{I} : \mathcal{A} \to \mathcal{P}(\mathcal{M}^0)$. We extend $\mathcal{I}$ to $\mathcal{T}$ (hence this includes $\mathbb{U}$) as follows:
• $\mathcal{I}(eU) = \mathcal{I}(U)^+$ • $\mathcal{I}(U \sqcap V) = \mathcal{I}(U) \cap \mathcal{I}(V)$ • $\mathcal{I}(U \to T) = \mathcal{I}(U) \rightsquigarrow \mathcal{I}(T)$
Because $\cap$ is commutative, associative, idempotent, and $(\mathcal{X} \cap \mathcal{Y})^+ = \mathcal{X}^+ \cap \mathcal{Y}^+$, $\mathcal{I}$ is well defined.

iv) Let $U \in \mathcal{T}$ (hence U can be in $\mathbb{U}$). We define the meaning $[U]$ of U by:
$[U] = \{M \in \mathcal{M} \mid M \text{ is closed and } M \in \bigcap_{\mathcal{I} \text{ interpretation}} \mathcal{I}(U)\}$.

It is easy to show that if $x^n \ N_1 ... N_k \in \mathcal{N}_x^n$ then $\forall \ 1 \leq i \leq k, \mathrm{d}(N_i) \geq n$.
Type interpretations are saturated and interpretations of good types contain only good terms.

4 The typing systems $\vdash_1$ and $\vdash_2$

In this section we introduce $\vdash_1$ and $\vdash_2$, our two intersection type systems with expansion variables. In $\vdash_1$, types are not restricted and SR fails. In $\vdash_2$, the syntax of types is restricted in the sense that arrows cannot occur to the left of intersections or expansions. In order to guarantee SR for this type system (and hence completeness later on), we introduce a subtyping relation which will allow intersection type elimination (something not available in the first type system).

Definition 8 Let $i \in \{1, 2\}$. The type system $\vdash_1$ (resp. $\vdash_2$) uses the set $\mathcal{T}$ (resp. $\mathbb{U}$) of definition 3. We follow [4] and write type judgements as $M : \langle \Gamma \vdash U \rangle$ instead of the traditional format of $\Gamma \vdash M : U$. The typing rules of $\vdash_i$ are (recall that when used for $\vdash_1$, U and T range over $\mathcal{T}$, and when used for $\vdash_2$, U ranges over $\mathbb{U}$ and T ranges over $\mathbb{T}$) of figure 4 (left). In the last clause, the binary relation $\sqsubseteq$ is defined on $\mathbb{U}$ by the rules of figure 4 (right).

Let Φ denote types in $\mathbb{U}$, or environments Γ or typings $\langle \Gamma \vdash_2 U \rangle$. When $\Phi \sqsubseteq \Phi'$, then Φ and Φ' belong to the same set ($\mathbb{U}$/environments/typings). Let Γ be an environment, $U \in \mathcal{T}$ and $M \in \mathcal{M}$.
• We say that Γ is $\vdash_i$-legal iff there are M, U such that $M : \langle \Gamma \vdash_i U \rangle$.
• We say that $\langle \Gamma \vdash_i U \rangle$ is good iff Γ and U are good.
• We say that $\mathrm{d}(\langle \Gamma \vdash_i U \rangle) > 0$ iff $\mathrm{d}(\Gamma) > 0$ and $\mathrm{d}(U) > 0$.

We show that typable terms are good, have good types, and have the same degree as their types and that all legal contexts are good. We also show that no β-redexes are blocked in a typable term.

SR for β using $\vdash_1$ fails: let a, b, c be different elements of $\mathcal{A}$. Although $(\lambda x^0.x^0x^0)(y^0z^0) \rhd_\beta (y^0z^0)(y^0z^0)$ and $(\lambda x^0.x^0x^0)(y^0z^0) : \langle y^0 : b \to ((a \to c) \sqcap a), z^0 : b \vdash_1 c\rangle$, it is not possible that $(y^0z^0)(y^0z^0) : \langle y^0 : b \to ((a \to c) \sqcap a), z^0 : b \vdash_1 c\rangle$.

Nevertheless, we show that SR and subject expansion for β using $\vdash_2$ holds. This will be used in the proof of completeness (more specifically in lemma 18 which is basic for the completeness theorem 19).

Lemma 9 (SUBJECT REDUCTION AND EXPANSION FOR β)
i) If $M : \langle \Gamma \vdash_2 U\rangle$ *and* $M \rhd^*_\beta N$, *then* $N : \langle \Gamma \vdash_2 U\rangle$.
ii) If $N : \langle \Gamma \vdash_2 U\rangle$ *and* $M \rhd^*_\beta N$ *then* $M : \langle \Gamma \vdash_2 U\rangle$.

The semantics given in section 3 is sound with respect to $\vdash_1$ and $\vdash_2$, because if $\mathcal{I}$ is an interpretation and $U \sqsubseteq V$ then $\mathcal{I}(U) \subseteq \mathcal{I}(V)$.

Lemma 10 (SOUNDNESS OF $\vdash_1/\vdash_2$) *Let* $i \in \{1, 2\}$, $\mathcal{I}$ *be an interpretation,* $M : \langle (x_j^{n_j} : U_j)_n \vdash_i U\rangle$ *and* $\forall 1 \leq j \leq n$, $N_j \in \mathcal{I}(U_j)$. *If* $M[(x_j^{n_j} := N_j)_n] \in \mathcal{M}$, *then* $M[(x_j^{n_j} := N_j)_n] \in \mathcal{I}(U)$.

Hence, if $M : \langle () \vdash_i U\rangle$, then $M \in [U]$. The next lemma puts the realisability semantics in use.

Lemma 11 *i)* $[(a \sqcap b) \to a] = \{M \in \mathbb{M}^0 \mid M \rhd^*_\beta \lambda y^0.y^0\}$.
ii) It is not possible that $\lambda y^0.y^0 : \langle () \vdash_1 (a \sqcap b) \to a\rangle$.
iii) $\lambda y^0.y^0 : \langle () \vdash_2 (a \sqcap b) \to a\rangle$.

Remark 12 (FAILURE OF COMPLETENESS FOR $\vdash_1$) Lemma 11 shows that we can not have a completeness result (a converse of lemma 10 for closed terms) for $\vdash_1$. To type the term $\lambda y^0.y^0$ by the type $(a \sqcap b) \to a$, we need an elimination rule for $\sqcap$ which we have in $\vdash_2$. However, we will see that we have completeness for $\vdash_2$ if only one expansion variable is used.

5 Completeness of $\vdash_2$ with one expansion variable

Let $a \in \mathcal{A}$, $e_1, e_2 \in \mathcal{E}$, $e_1 \neq e_2$ and $Nat_0 = (e_1a \to a) \to (e_2a \to a)$. Then:
1) $\lambda f^0.f^0 \in [Nat_0]$ and 2) It is not possible that $\lambda f^0.f^0 : \langle () \vdash_2 Nat_0\rangle$.

Hence $\lambda f^0.f^0 \in [Nat_0]$ but $\lambda f^0.f^0$ is not typable by Nat_0 and we do not have completeness in the presence of more than one expansion variable. The problem comes from the fact that for the realisability semantics that we considered, we identify all expansion variables. In order to give a completeness theorem we will in what follows restrict our system to only one expansion variable. In the rest of this section, we assume that the set $\mathcal{E}$ contains only one expansion variable e_c.

The need of one single expansion variable is clear in part 2) of lemma 13 which would fail if we use more than one expansion variable. For example, if $e_1 \neq e_2$ then $e_1(e_2a)^- = e_1a \neq e_2a$. This lemma is crucial for the rest of this section and hence, a single expansion variable is also crucial.

Lemma 13 *Let* $U, V \in \mathbb{U}$ *and* $\mathrm{d}(U) = \mathrm{d}(V) > 0$. *1)* $e_cU^- = U$ *and 2) If* $U^- = V^-$, *then* $U = V$.

Next, we divide $\{y^n \mid y \in \mathcal{V}_2\}$ disjointly amongst types of order n.

Definition 14 Let $U \in \mathbb{U}$. We define sets of variables $\mathbb{V}_U$ by induction on $\mathrm{d}(U)$. If $\mathrm{d}(U) = 0$, then: $\mathbb{V}_U$ is an infinite set of variables of degree 0; if $y^0 \in \mathbb{V}_U$, then $y \in \mathcal{V}_2$; and if $U \neq V$ and $\mathrm{d}(U) = \mathrm{d}(V) = 0$, then $\mathbb{V}_U \cap \mathbb{V}_V = \emptyset$. If $\mathrm{d}(U) = n + 1$, then we put $\mathbb{V}_U = \{y^{n+1} \mid y^n \in \mathbb{V}_{U^-}\}$.

Our partition of $\mathcal{V}_2$ allows useful infinite sets which contain type environments that will play a crucial role in one particular type interpretation. These sets and environments are given in the next definition.

Definition 15 *i)* Let $n \in \mathbb{N}$. We let $\mathbb{G}^n = \{(y^n : U) \mid U \in \mathbb{U}, \mathrm{d}(U) = n \text{ and } y^n \in \mathbb{V}_U\}$ and $\mathbb{H}^n = \bigcup_{m \geq n} \mathbb{G}^m$. Note that $\mathbb{G}^n$ and $\mathbb{H}^n$ are not type environments because they are infinite sets.
ii) Let $n \in \mathbb{N}$, $M \in \mathcal{M}$ and $U \in \mathbb{U}$, we write $M : \langle \mathbb{H}^n \vdash_2 U \rangle$ iff there is a type environment $\Gamma \subset \mathbb{H}^n$ where $M : \langle \Gamma \vdash_2 U \rangle$

Now, for every n, we define the set of the good terms of order n which contain some free variable x^i where $x \in \mathcal{V}_1$ and $i \geq n$.

Definition 16 Let $n \in \mathbb{N}$ and $\mathcal{V}^n = \{M \in \mathbb{M}^n \mid x^i \in FV(M) \text{ where } x \in \mathcal{V}_1 \text{ and } i \geq n\}$. Obviously, if $n \in \mathbb{N}$ and $x \in \mathcal{V}_1$, then $\mathcal{N}^n_x \subseteq \mathcal{V}^n$.

Here is the crucial interpretation $\mathbb{I}$ for the proof of completeness:

Definition 17 Let $\mathbb{I}$ be the interpretation defined by:
for all type variables a, $\mathbb{I}(a) = \mathcal{V}^0 \cup \{M \in \mathcal{M}^0 \mid M : \langle \mathbb{H}^0 \vdash_2 a \rangle\}$.

$\mathbb{I}$ is indeed an interpretation and the interpretation of a type of order n contains the good terms of order n which are typable in the special environments which are parts of the infinite sets of definition 15:

Lemma 18 *i) $\mathbb{I}$ is an interpretation. I.e., $\forall a \in \mathcal{A}$, $\mathbb{I}(a)$ is saturated and $\forall x \in \mathcal{V}_1, \mathcal{N}^0_x \subseteq \mathbb{I}(a) \subseteq \mathbb{M}^0$.*
ii) If $U \in \mathbb{U}$ is good and $\mathrm{d}(U) = n$, then $\mathbb{I}(U) = \mathcal{V}^n \cup \{M \in \mathbb{M}^n \mid M : \langle \mathbb{H}^n \vdash_2 U \rangle\}$.

$\mathbb{I}$ is used to prove completeness (the proof is on the authors web pages).

Theorem 19 (COMPLETENESS) *Let $U \in \mathbb{U}$ be good such that $\mathrm{d}(U) = n$.*
i) $[U] = \{M \in \mathbb{M}^n \mid M : \langle () \vdash_2 U \rangle\}$.
ii) $[U]$ is stable by reduction: i.e., if $M \in [U]$ and $M \rhd^*_\beta N$, then $N \in [U]$.
iii) $[U]$ is stable by expansion: i.e., if $N \in [U]$ and $M \rhd^*_\beta N$, then $M \in [U]$.

6 Conclusion and future work

We studied the $\lambda I^{\mathbb{N}}$-calculus, an indexed version of the λI-calculus. This indexed version was typed using first an intersection type system with expansion variables but without an intersection elimination rule, and then using an intersection type system with expansion variables and an elimination rule.

We gave a realisability semantics for both type systems showing that the first type system is not complete in the sense that there are types whose semantic meaning is not the set of $\lambda I^{\mathbb{N}}$-terms having this type. In particular, we showed that $\lambda y^0.y^0$ is in the semantic meaning of $(a \sqcap b) \to a$ but it is not possible to give $\lambda y^0.y^0$ the type $(a \sqcap b) \to a$. The main reason for the failure of completeness in the first system is associated with the failure of the subject reduction property for this first system. We showed that the second system has the desirable properties of subject reduction and expansion and strong normalisation but that completeness fails if we use more than one expansion variable. We then showed that completeness succeeds if we restrict the system to one single expansion variable.

Because we show in the appendixes of the long version of this article (which can be downloaded on the web page of the authors) that each of these type systems, when restricted to the normal λI-calculus represents a well known intersection type system with expansion variables, our study can be said to be the first denotational semantics study of intersection type systems with expansion variables (using realisability or any other approach) and outlines the difficulties of doing so. Although we have in this

paper limited the study to the λI-calculus, future work will include extending this work to the full λ-calculus and with an ω-type rule as well.

References

[1] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, revised edition, 1984.

[2] H. P. Barendregt. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, T. S. E. Maibaum, eds., *Handbook of Logic in Computer Science*, vol. 2, chapter 2. Oxford University Press, 1992.

[3] S. Carlier, J. Polakow, J. B. Wells, A. J. Kfoury. System E: Expansion variables for flexible typing with linear and non-linear types and intersection types. In *Programming Languages & Systems, 13th European Symp. Programming*, vol. 2986 of *LNCS*. Springer-Verlag, 2004.

[4] S. Carlier, J. B. Wells. Expansion: the crucial mechanism for type inference with intersection types: A survey and explanation. In *Proc. 3rd Int'l Workshop Intersection Types & Related Systems (ITRS 2004)*, 2005. The ITRS '04 proceedings appears as vol. 136 (2005-07-19) of *Elec. Notes in Theoret. Comp. Sci.*

[5] M. Coppo, M. Dezani-Ciancaglini, B. Venneri. Principal type schemes and λ-calculus semantics. In J. R. Hindley, J. P. Seldin, eds., *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*. Academic Press, 1980.

[6] T. Coquand. Completeness theorems and lambda-calculus. In P. Urzyczyn, ed., *TLCA*, vol. 3461 of *Lecture Notes in Computer Science*. Springer, 2005.

[7] S. Farkh, K. Nour. Résultats de complétude pour des classes de types du système AF2. *Theoretical Informatics and Applications*, 31(6), 1998.

[8] G. Goos, J. Hartmanis, eds. *λ-Calculus and Computer Science Theory, Proceedings of the Symposium Held in Rome, March 15-27, 1975*, vol. 37 of *Lecture Notes in Computer Science*. Springer-Verlag, 1975.

[9] J. R. Hindley. The simple semantics for Coppo-Dezani-Sallé types. In M. Dezani-Ciancaglini, U. Montanari, eds., *International Symposium on Programming, 5th Colloquium*, vol. 137 of *LNCS*, Turin, 1982. Springer-Verlag.

[10] J. R. Hindley. The completeness theorem for typing λ-terms. *Theoretical Computer Science*, 22, 1983.

[11] J. R. Hindley. Curry's types are complete with respect to F-semantics too. *Theoretical Computer Science*, 22, 1983.

[12] J. R. Hindley. *Basic Simple Type Theory*, vol. 42 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1997.

[13] F. Kamareddine, K. Nour. A completeness result for a realisability semantics for an intersection type system. *Annal of Pure and Applied Logic*, 146, 2007.

[14] J. Krivine. *Lambda-Calcul : Types et Modèles*. Etudes et Recherches en Informatique. Masson, 1990.

[15] R. Labib-Sami. Typer avec (ou sans) types auxilières.